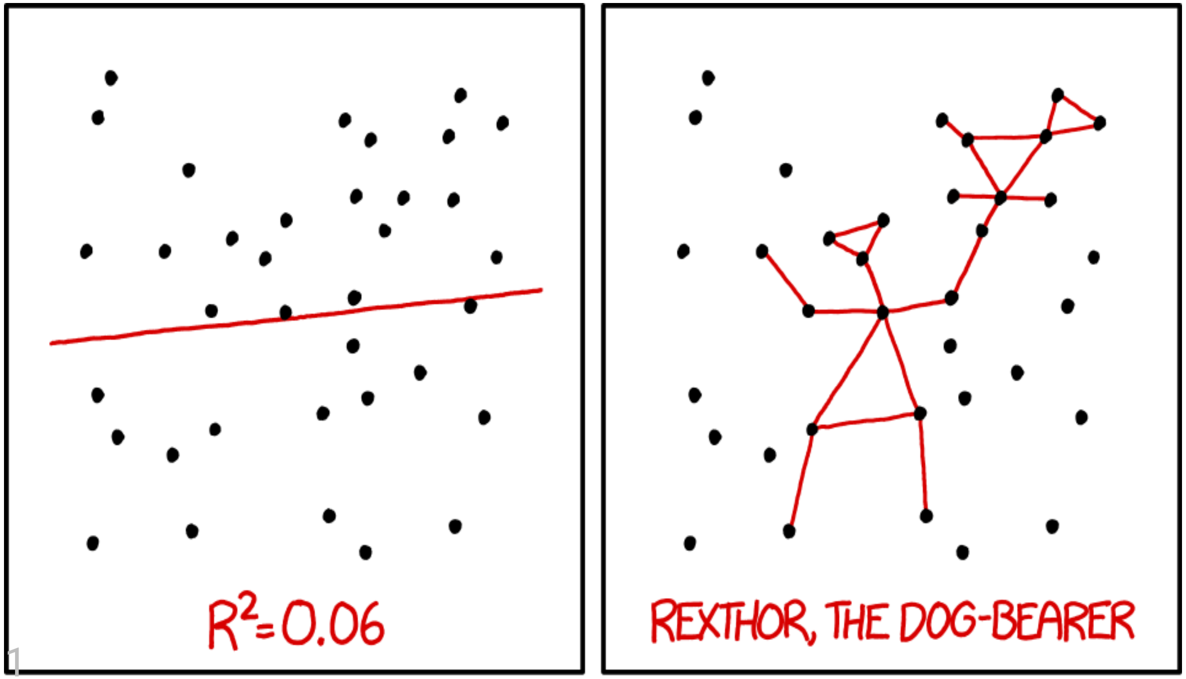
# Stat 602 Final Project:
# Don't Overfit! II

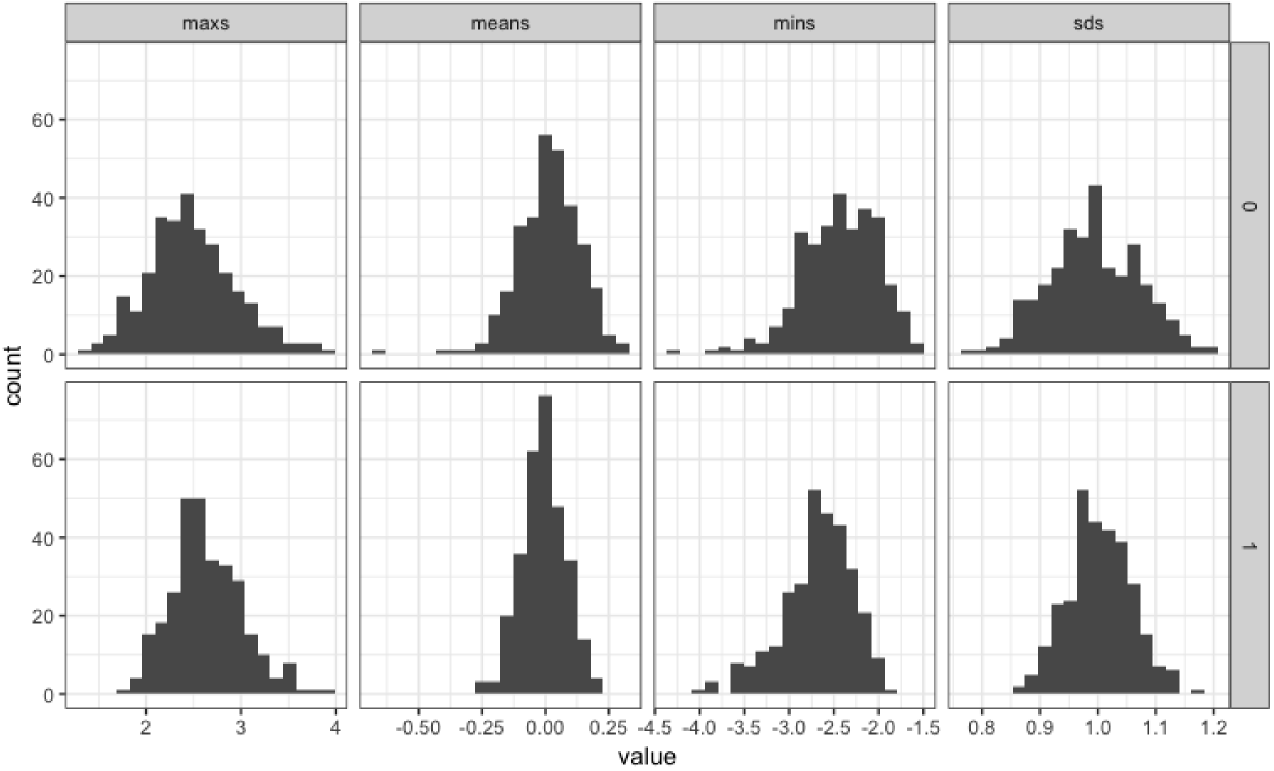
Nate Garton and Kiegan Rice



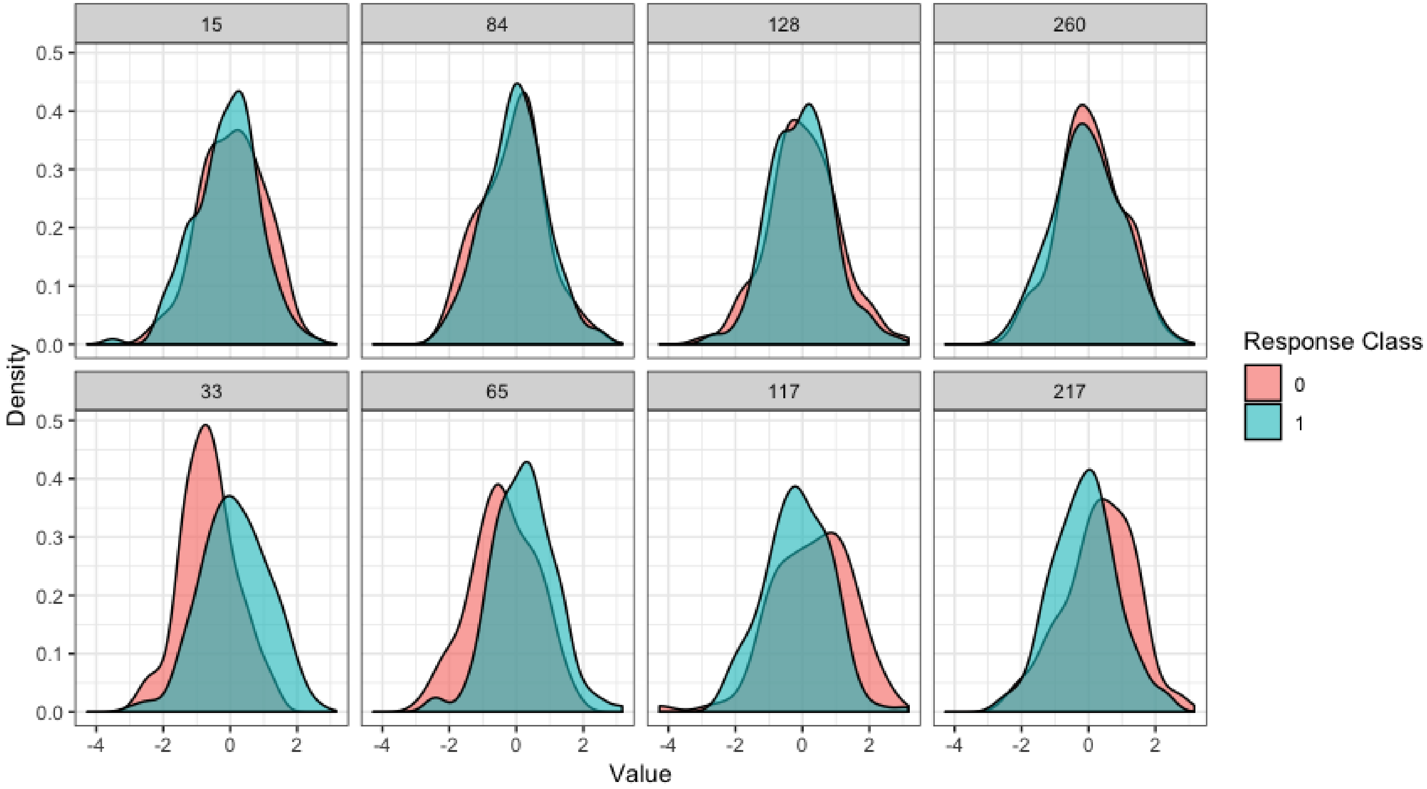
# Background

- Kaggle’s “Don’t Overfit! II” competition
- 250 cases in training / 19,750 in testing
    - 1,975 (10%) testing cases used for public leaderboard
- AUROC used for scoring
- 300 feature columns
    - no column names
- Response: 0-1 Classification

# EDA


maxs
means
mins
sds
0
1
count
value
0
20
40
60
2
3
4
-0.50
-0.25
0.00
0.25
-4.5
-4.0
-3.5
-3.0
-2.5
-2.0
-1.5
0.8
0.9
1.0
1.1
1.2

# EDA: Class Conditional Densities



# EDA: Class Conditional Densities

- Shapiro-Wilk tests for normality for each column conditional on class
- KS tests that p-values from SW test are $U(0,1)$
  - p-value for class 0: $0.904$
  - p-value for class 1: $0.997$
- Fisher transformation on class conditional column correlations
- KS tests that p-values from Z-tests are $U(0,1)$
  - p-value for class 0: $0.113$
  - p-value for class 1: $0.905$
- t-tests of differences in column means between classes
  - columns 33 and 65 have p-values that survive Bonferroni correction

# Feature Engineering

- Only consider functions of variables retained by a LASSO on raw data
- univariate Gaussian log-LRs
- univariate kernel density log-LRs

# Model Fitting

- Random forest with all raw features + Gaussian log-LRs
    - using OOB error for training
    - Final AUROC: 0.773 ($\downarrow$ 0.007)
- LASSO with all raw features + Gaussian log-LRs ($\lambda_{1se}$)
    - keeps only the raw column 33 and 65
    - Final AUROC: 0.833 ($\downarrow$ 0.015)
- Combined predictions for a few models
    - Averaged testing set predictions on two LASSO models and a random forest
    - (Barely) better public AUROC initially
    - Final AUROC: 0.833 ($\downarrow$ 0.016)

# Failed Ideas

- Fitting flexible predictors on all raw data (or raw data + engineered features)
  - Fitting flexible predictors on small sets of engineered features worked better, but not competitive
- Unsupervised clustering to hunt for signal, relationships between variables
  - Model-based clustering using `Mclust` package
  - Resulting clusters essentially random sample of response classes

# Takeaways

- Feature engineering is difficult when there is:
    - small sample size
    - no context for features or response
- Assumptions are important, but CV is king
- Food for thought:
    - With LASSO, performance doesn’t seem to suffer when adding more predictors. With more “flexible” predictors, this is not the case even though we still use CV to choose penalty parameters. Why?
- We did not win

# If we had more time…

- Test several hypothetical generative data models for compatibility with training data
- Bayes
- Testing additional classification methods
- True “ensembling” with one feature matrix and multiple prediction methods